KIP-49 - Fair Partition Assignment Strategy

Status

Current state: Under Discussion

Discussion thread: Mailing list thread link

JIRA:

- KAFKA-2435
- KAFKA-3297

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

While the roundrobin partition assignment strategy is generally a substantial improvement over the original range strategy, when the consumer topic subscriptions are not identical it can generate heavily skewed assignments.

Because of that known deficiency, the roundrobin strategy initially disallowed non-identical subscriptions. This was found to be too restrictive of a constraint in practice. For example, in a typical highly available production deployment the members of a consumer group may be restarted in rolling fashion when their subscription configuration changes. The group would be unable to consume messages for the duration of the rolling update. A similar situation arises if each member of the group were to dynamically refresh its topic subscription list on a regular interval. Depending on the refresh interval duration and synchronicity across members, a topic change could be very disruptive, causing consumption to completely stall with a large message lag possibly accumulating.

Consequently KAFKA-2172 proposed removing this restriction.

Kafka currently lacks an alternative strategy that attempts to assign an equal number of partitions to each consumer in a group, regardless of how similar their individual topic subscriptions are.

Public Interfaces

This proposal only affects client configuration options.

- For the original high-level consumer, add a "fair" value for the "partition.assignment.strategy" client configuration, so that the valid values would be "range", "roundrobin", and "fair".
- For the new consumer, add a FairAssignor class than can be selected by setting the "partition.assignment.strategy" client configuration to "org.apache.kafka.clients.consumer.FairAssignor".

The default value for "partition.assignment.strategy" would remain unchanged for both the original high-level consumer and the new consumer.

Proposed Changes

Provide a Fair Partition Assignor option for both the original high-level consumer and the new consumer.

This assignor would use a different algorithm than the roundrobin assignor, tracking the number of partitions assigned to each consumer group member, and having the partition assignment assign each partition to the least-loaded eligible consumer.

Additionally, it has been found that we can optimize the distribution fairness by adjusting the partition assignment order:

- Topics with fewest consumers are assigned first.
- In the event of a tie for fewest consumers, the topic with more partitions is assigned first.

The general idea behind these two rules is to keep the most flexible assignment choices available as long as possible by starting with the most constrained partitions / consumers.

Example

The following example demonstrates how this new algorithm could be beneficial.

Given the following inventory of five topics,

Topic	Partitions
T1	2
T2	1
T3	2
T4	1
T5	2

for a total of eight partitions, and a consumer group consisting of four members (C1, C2, C3, C4) with the following topic subscriptions,

Consumer	Topics
C1	T1, T2, T3, T4, T5
C2	T1, T3, T5
C3	T1, T3, T5
C4	T1, T2, T3, T4, T5

Note that this sort of situation could plausibly occur after an initial state of all consumers subscribing to (T1, T3, T5) in the window of time from when the configuration for C1 and C4 has been refreshed to consume the two additional topics, until C2 and C3 are eventually updated some time later.

here are the resulting assignments for each of the assignment strategies,

Range

Consumer	Assigned Partitions
C1	T1-0, T2-0, T3-0, T4-0, T5-0
C2	T1-1, T3-1, T5-1
C3	
C4	

Round Robin

Consumer	Assigned Partitions
C1	T1-0, T3-0, T5-0
C2	T1-1, T3-1, T5-1
C3	
C4	T2-0, T4-0

Fair

Consumer	Assigned Partitions
C1	T2-0, T3-0
C2	T1-0, T3-1
C3	T1-1, T5-0
C4	T4-0, T5-1

Compatibility, Deprecation, and Migration Plan

This would be a passive (opt-in) change. No existing functionality will be deprecated.

Rejected Alternatives

The roundrobin strategy for the original high-level consumer initially disallowed non-identical subscriptions among members of a consumer group. As discussed in KAFKA-2172 and above this is problematic for rolling configuration changes. When that Jira has been completed and non-identical subscriptions are allowed for the roundrobin strategy, we can document that this leniency may produce significantly unbalanced assignments in some cases. It seems inadequate to address this design flaw in the roundrobin strategy with known-issue documentation only, and inappropriate to redesign the existing roundrobin strategy to be fairness-aware for unbalanced subscriptions.

For the new consumer a custom implementation class can potentially be configured rather than adding a third built-in option into Kafka. As the logic is somewhat complicated to get correct, it likely makes sense for Kafka to provide this as a convenience.

It has been noted that with heavily skewed subscriptions, fairness is sort of moot – i.e., people would generally scale up or down subscription counts with the express purpose of reducing/increasing load on those instances. While a valid point, that would not be true in the case of rolling configuration updates as described above.